Research Article
i-Proclaim .my
Asia Pac. j. energy environ.

Hybrid Automata: An Algorithmic Approach Behavioral Hybrid Systems

Venkata Koteswara Rao Ballamudi

Solution Architect, Software Engineering Architecture Department, Cyber Global, Inc., **USA**

*Email for Correspondence: koteswarabv@gmail.com

Abstract

Hybrid automata strategies have advanced as a vital tool to design, check and direct the execution of hybrid systems. Any way they can – and we assume should – be utilized to communicate quantitative models about hybrid systems in different areas, for example, experimental sciences. Since the conventional design of hybrid automata compares well to consecutively integrate behavioral chains in living creatures, we look for a use of hybrid modeling procedures in the social sciences and, particularly, brain research. We attempt to address the question related to how human drivers move onto an expressway and simultaneously utilize this study as our test-bed for utilizing hybrid automata inside behavioral sciences. Hybrid automata give a language to displaying and exploring advanced and simple calculations in real-time systems. Hybrid automata are studied here from a dynamical systems point of view. Essential and adequate conditions for the presence and uniqueness of arrangements are inferred and a class of hybrid automata whose arrangements rely consistently upon the underlying state is described. The outcomes on presence, uniqueness, and progression fill in as a beginning stage for solid study. In this paper, we present the structure of hybrid automata as a model and detailed language for hybrid systems. Hybrid automata can be seen as a theory of timed automata, in which the behavior of factors is represented in each state by a bunch of differential conditions. We show that a large number of the models considered in the workshop can be characterized by hybrid automata. While the reachability issue is undecidable in any event, for extremely confined classes of hybrid automata, we present two semi-decision techniques for checking security properties of piecewise-straight hybrid automata, in which all factors change at steady rates. The two techniques are based, individually, on limiting and figuring fix points on for the most part endless state spaces. We show that if the end of the method, at that point they offer the right responses. We then show that for a significant number of the run of the mill workshop models, the strategies do end and hence give an algorithmic approach to confirming their properties.

Key words

Hybrid automata, Real-time systems, Theory of timed automata, Algorithmic approach

9/20/2019 **Source of Support:** None | **No Conflict of Interest:** Declared

Introduction

A hybrid system is a dynamic system with both discrete and continuous sections. For instance, an automobile engine whose fuel infusion (constant) is managed by a microchip (discrete) is a hybrid system. As embedded computing becomes universal, hybrid systems are progressively utilized in security basic applications, hence making unwavering quality a superb concern. Thorough reliability examination requires formal modeling. For this reason, hybrid automata have been proposed as a proper model for hybrid systems.

Increasingly more real-life processes, from lifts to airplanes, are constrained by programs. These receptive projects are inserted in consistently evolving conditions and should respond to environmental changes continuously. Accuracy is of indispensable significance for responsive projects. However customary program check strategies permit us, best-case scenario, to estimated consistently evolving conditions by discrete analysis. A summed up conventional model for registering systems is expected to loyally address both discrete and continuous projects inside an integrated structure. Hybrid automata present such a structure.

A hybrid system comprises a discrete program inside a simple climate. Hybrid automata are generalized limited state machines for modeling hybrid systems. Of course, the discrete advances of a program are modeled by a difference in the program counter, which ranges over a limited arrangement of control areas. Also, we take into consideration the likelihood that the worldwide condition of a system changes consistently with time as indicated by the laws of physical science. For each control area, the ceaseless exercises of the climate are represented by a bunch of differential conditions. We name every area additionally with an invariant condition that should hold while the control lives at the area, and each change is named with a watched set of tasks.

Hybrid automata have been created as a formalism to model systems in which discrete control rationale collaborates with a genuine esteemed reality and encourage numerical verifications about their social properties. The discrete control rationale as a rule conceptualized is silicon-based, yet hybrid automata make no such assumption. Hybrid automata are traditionally utilized for modeling hybrid systems and their programming in various fields of specialized designing. The emphasis lies on the dedication of such systems and consequently on the conventional check of properties. A few model checkers (Henzinger et al., 1997; Frehse, 2005; Taher-Uz-Zaman et al., 2014) have been created for this reason. Other than their conventional semantics, hybrid automata offer satisfying visual documentation open with just at least formal training.

TRACE LANGUAGES OF HYBRID AUTOMATA

We recognize which prerequisites on the traces of a hybrid machine can be checked algorithmically, and which can't.

Verification Tasks

We study four paradigmatic questions regarding the hints of hybrid automation. The reachability issue is a central subtask for the verification of security needs, and the vacancy issue is a crucial subtask for the verification of liveness prerequisites. The timed trace follows a consideration issue as compared to the hints of a hybrid machine against a coordinated specification, and the time abstract trace inclusion problem compares the traces of hybrid automata against a time-abstract specification.

Rectangular Automata

Hybrid automation is rectangular if the stream conditions are free of the control modes, and the factors are pairwise autonomous. Specifically, in each control method of rectangular automata, the first subordinate of every factor is given a scope of potential qualities, and that reach doesn't change with control switches. With each control switch of rectangular automata, the estimation of every factor is either left unchanged or changed non-deterministically to another value inside a given scope of possible outcomes. The practices of the factors are decoupled because the scopes of potential qualities and subordinate qualities for one variable can't rely upon the worth or subordinate estimation of another variable.

HYBRID MODELING OF BEHAVIORAL VARIABLES

This segment presents some variable systems of modeling behavior and its formalization regarding hybrid automata, represented by rather simple instances of behavior.

Hybrid Automata as a formalism for the behavior of organisms

The behavior of organic entities can be organized as an order of various behavioral systems and subsystems (see Figure 1). Depending upon inner states and outer impacts the organic entity may change starting with one behavior system then onto the next. The controlling variable of such behavioral changes is believed to be an interior portrayal of all relevant situational factors, subjective by their predicted results – the alleged reinforcement value. The reinforcement value can be considered as a "typical currency" by which all predicted results of behavior are assessed. Switching starting with one social state then onto the next may happen inside a single system. An illustrative model is alleged Sequential Action Patterns (SAPs). SAPs are fixed social sequences, which are inspired by a naturally significant upgrade (Möhr, 1989). In each condition of the arrangement, there is a bunch of factors that control the behavior of the creature in a certain way. The impact of the controlling factors in a specific system state could – on a basic level – be indicated by a bunch of differential conditions. If specific factors arrive at a basic value, notwithstanding, the life form changes to another state in which a diverse arrangement of factors may control the behavior (Donepudi, 2017). It might likewise happen that the same factors impact the behavior in an alternate manner. The change from one behavioral state to another inside an SAP may accordingly be communicated as a discrete change starting with one bunch of differential conditions then onto the next.

We will demonstrate an SAP as a hybrid automata by addressing each discrete condition of the life form as a control condition of the automation powered by hybrid systems. The expected factors of the life form are addressed by factors in the machine and the differential conditions running the factors of the creature are related to control states as shown by the mapping from behavioral state to control state.

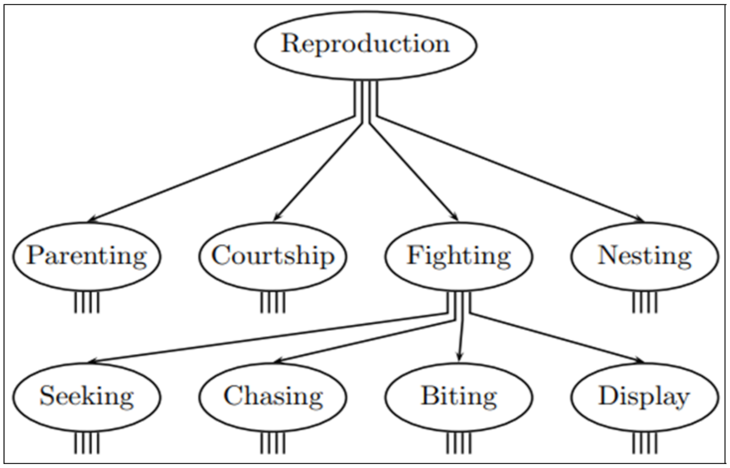

Figure 1: Organization of Reproductive Behavioral Systems

Some examples of SAPs

The capability of this methodology will be represented in this section by a couple of models for consecutively coordinated behavior in creatures and people.

Mating

The three-spined stickleback is a species of little fish with somewhat perplexing mating behavior, which is coordinated in three different stages. To begin with, the male builds up a domain and develops a home by diving an opening in the ground and gathering soft material. In this stage, the presence of a female doesn't have any impact on the male's behavior. When the home is done, be that as it may, females entering the domain are sought and incited to lay eggs in the home. Once the eggs are laid the presence of female capacities as a trigger to pursue it away.

From this textual portrayal, one can determine automation like Figure 2. Modeling the volume of the opening hole so far by the machine begins in the state "Build" where the volume of the opening is expanded consistently (Donepudi, 2015). When the opening has arrived at the ideal size (the automata needs to make the limit specific), seeking behavior starts. This model expects the presence of the closest female, the d variable following its distance to the home. During courting, the distance is brought down until it arrives at nothing, so, all things considered, the female is chased away.

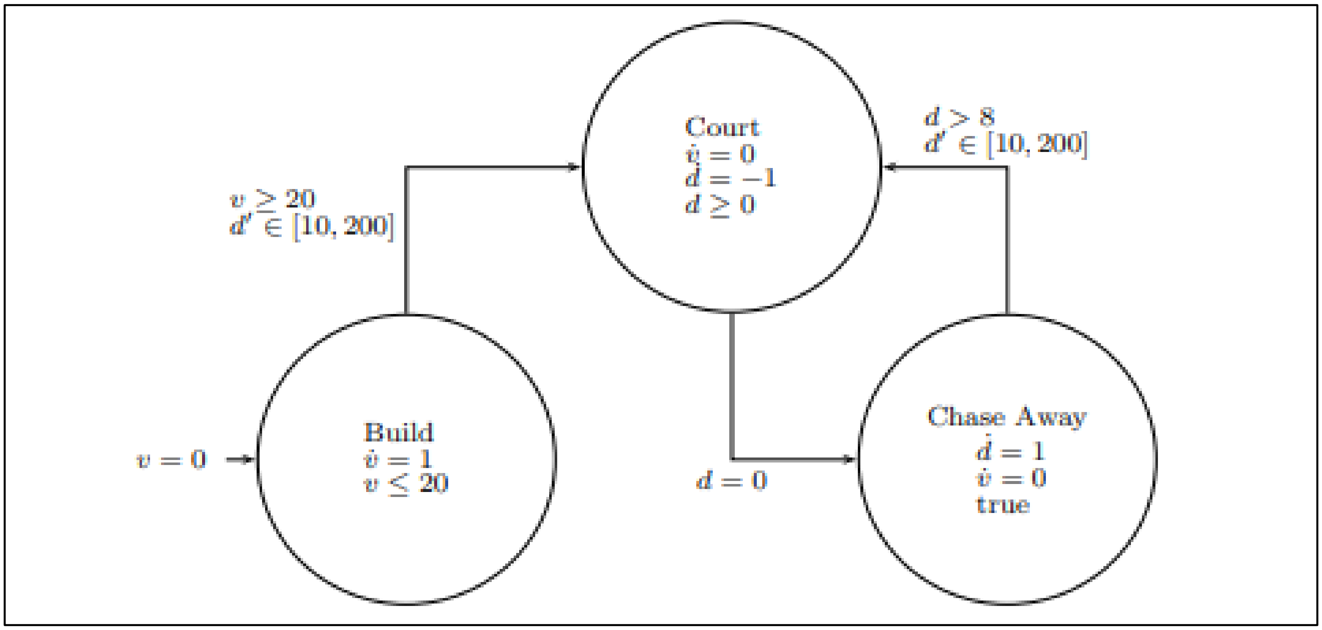

Figure 2: A model for courting behavior of three-spined stickleback

Easter Egg Search

Like the qualification among global and focal search mode in animal foraging, people additionally some of the time switches between various procedures while executing a single task. Consider the yearly assignment of discovering easter eggs covered up inside some very much characterized region. The exchanging of procedures can be depicted by the hybrid machine shown in Figure 3. At first, one outputs the region, attempting to search for areas that show up to ensure great disguise. Having recognized such an area, one moves there also, attempts to discover an egg in that (which may not, in reality, exist, as displayed by egg). The activities displayed during the two periods of search are extraordinary.

Note that during the global search mode the event of an easter egg has no impact on the searcher's conduct. This implies that one doesn't notice an egg on the off chance that he/she is searching for potential concealing spots. Since Easter egg search has once in a while been explored systematically until the current day, this forecast stays theoretical – showing the capability of hybrid modeling to produce testable speculations. Most people likewise note the decrease in search achievement once most eggs have been found and finally abandon the search. We model this by monitoring the time during which no egg was found.

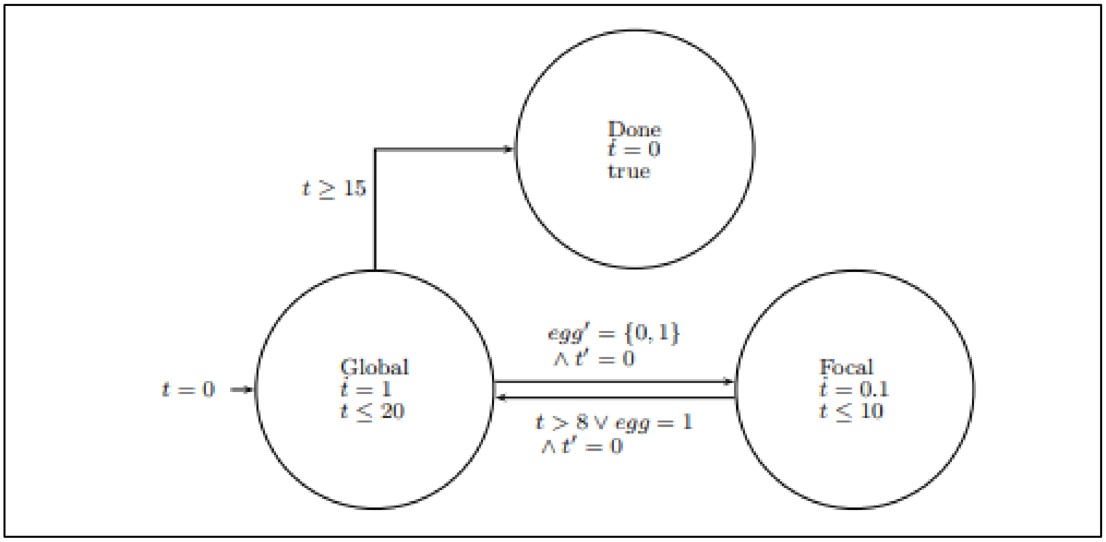

Figure 3: A model for easter egg search

APPLICATIONS OF HYBRID AUTOMATA

Temperature regulator

Our first application depicts a nonlinear hybrid system. The temperature of a plant is controlled through an indoor regulator, which ceaselessly facilities the temperature and turns a radiator on and off. The temperature is represented by differential conditions.

At the point when the radiator is off, the temperature, meant by the variable x, reduces as indicated by the remarkable capacity z(t) = He-Kt, where t is the time, 0 is the underlying temperature, and K is a consistent dictated by the plant; when the heater is on, the temperature follows the capacity z(t) = de-Kt + h(1 - e-Kt), where 'h' is a constant that relies upon the force of the heater. Assume that at first, the temperature is M degrees and the heater is turned off. We wish to keep the temperature between m and M degrees. The subsequent system can be depicted by the hybrid automata of Figure 4 (note the representation of the underlying condition x = M). The automata have two locations: in area l_0, the heater is turned off; in area l_1, the heater is on.

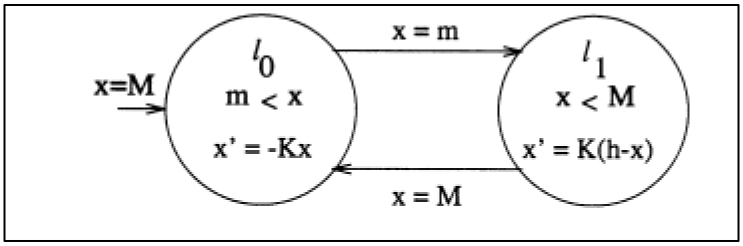

Figure 4: Automata of Temperature Controller

Water Level Monitor

Our next application depicts a linear system. The water level in a tank is controlled through a monitor, which consistently looks over the water level and turns a pump on and off. In contrast to the temperature, the water level changes as a piecewise-linear capacity over the long haul. At the point when the pump is off, the water level, denoted by the variable y, falls by 2 inches every second; when the pump is on, the water level increases by 1 inch for each second. Assume that at first, the water level is 1 inch and the pump is turned on. We wish to keep the water somewhere in the range of 1 and 12 inches. However, from the time that the monitor signs to change the status of the pump to the time that the change gets viable, there is a delay of 2 seconds. Accordingly, the monitor should signal toward turn the pump on before the water level falls to 1 inch, and it should signal toward turning the pump off before the water level arrives at 12 inches.

The linear hybrid machine of Figure 5 portrays a water level monitor that signals at whatever point the water level passes 5 and 10 inches, individually. The automation has four locations: in areas l_0 and l_1, the pump is turned on; in areas l_2 and l_3, the pump is off. The clock x is utilized to indicate the delays: at whatever point the automation control is in area l_1 or l_3, the sign to turn the pump off or on, separately, was sent x seconds back.

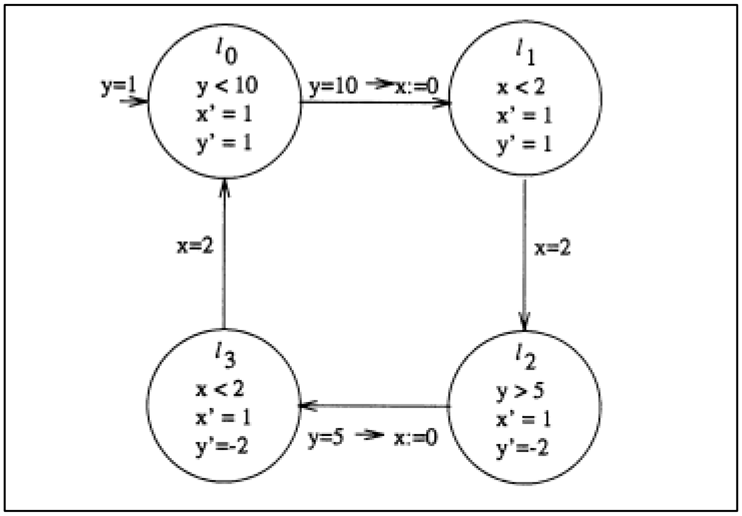

Figure 5: Water Level Monitor

Mutual-exclusion Protocol

This application portrays a defined multi-rate coordinated system. It has a timing-based calculation that executes mutual exclusion for a distributed system with smart timers. Consider an asynchronous shared-memory system that comprises two cycles P1 and P2 with atomic read and write operation. Each interaction has a basic location and at each time, generally, one of the two cycles is allowed to be in its basic location. Mutual exclusion in this scenario is guaranteed by a version of Fischer's protocol. For each cycle P_i, where i = 1, 2:

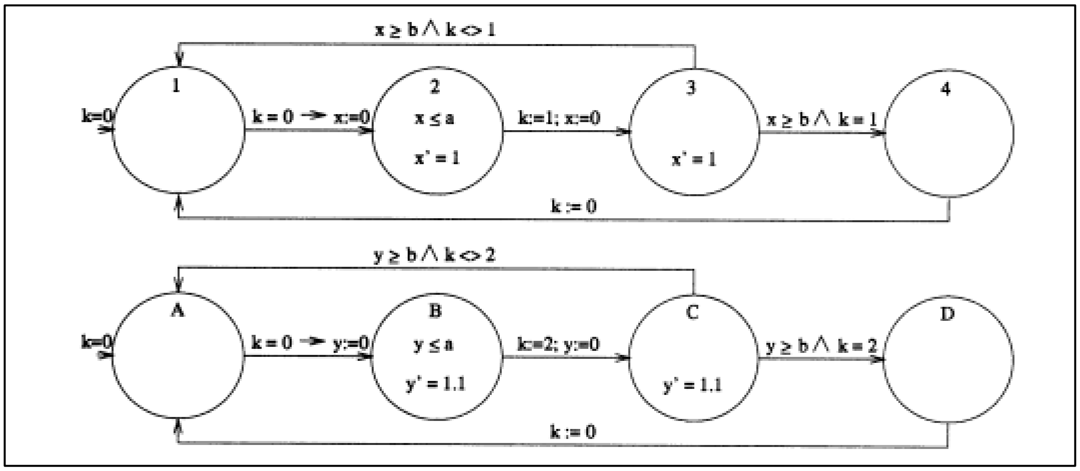

Figure 6: Mutual Exclusion Protocol

The two cycles P1 and P2 share a variable k and process P_i is allowed to be in its basic location if k = 1. Each cycle has a private clock. The instruction delay 'b' defers a cycle for at any rate 'b' time units as estimated by the process's local clock. Besides, each process takes at most a times units, as estimated by the cycle's clock, for a single writing operation to the shared memory (i.e., for the task k = 1). The estimations of 'a' and 'b' are the only data we have about the timing behavior of directions. The protocol guarantees mutual exclusion just for specific estimations of 'a' and 'b'. If both private processor times continue at exactly a similar rate, at that point mutual exclusion is ensured if a < b.

To make the application more intriguing, we assume that the two private timers of the cycles P1 and P2 continue at various rates, specifically, the local clock of P2 is 1.1 times quicker than the clock of P1. The resulting system can be demonstrated by the result of the two-hybrid automata introduced in Figure 6 above.

Hybrid Automation Simulation of Human Drivers

As effectively noted, we will attempt to utilize hybrid automata modeling to address questions about human life and death, or, in other words, behavior in rush hour traffic. For an individual who has simply figured out how to drive a vehicle, conscious intellectual processes decide a large portion of their driving behavior. As experience is referred to, in any case, the behavior will in general become computerized to a significant degree. Along these lines, controlling, slowing down and even path changes occur without the driver's consciousness being fundamental for controlling it. Both for the design of advanced driver assistance systems and the design of the driving environment it is helpful to execute this reality in a driver model. We took this issue as our testbed for the appropriateness of hybrid automata to the modeling of human behavior. As an initial step of the study, we limit our consideration regarding the circumstance of driving onto a highway.

Models of the behavior of human drivers are generally arranged in the domain of psychological brain research. Alleged intellectual driver models treat the driver's behavior because of some mind-boggling data handling techniques, which can be separated into a few phases of perceiving, assessing, objective setting, and choosing (Möbus et al., 2009; Salvucci et al., 2001; Donepudi, 2014b). A formalization of such models by hybrid automata began, yet ended up being very unpredictable. Psychological models, nonetheless, experience the ill effects of some serious issues: Firstly they model naturally undetectable processes. Also, furthermore, they will turn out to be exceptionally complicated, in any event, when presenting simple circumstances. We along these lines ground our model on noticeable behavior, to be specific direction and speed, restricting ourselves to just a single "hidden variable" – the reinforcement value. Our outcomes will show that, for our setting, it is possible to demonstrate such complex behavior consistently by a fairly basic model. We are sure that the methodology scales for even considerably more complex circumstances, for example, circumstances thinking about a few interacting drivers.

The methodology set forward in this paper is drastically unique to conventional, psychological models of driving behavior. As a beginning stage, we take the presumption that most driving behavior is an aftereffect of reinforcement value put into learning processes such as operant and traditional conditioning. It seems reasonable, accordingly, to hypothesize some sort of reinforcement value of important boundaries of the circumstance being referred to. By reinforcement value we mean the add up to which a result is more desirable over potential choices ("to slam into another vehicle", for instance, is less ideal than "to reach traveling speed"). This reinforcement value needn't bother with to be addressed consciously in the organism. Neither does it refer to a set objective, towards which the behavior of the whole organism is driven. The reinforcement value in a given circumstance is a (simply hypothetical) capacity of the predicted results of a potential behavioral pattern. It might rely upon natural variables (hereditary inclinations), as well as on the human's learning history. With regards to gained practices, for example, driving, the reinforcement value can be taken to be the aftereffect of the results of past driving conduct. Sudden changes in direction, for instance, are related with a negative value – not because of an evaluative interaction happening in the driver – yet rather due to (possibly risky) past results of unexpected direction change. By methods for the referenced feedback learning measures the behavior ultimately approximates an ideal answer for a given circumstance. Accordingly, our primary hypothesis is that drivers will in general carry on so that the value in a given circumstance is increased.

To imitate the improving process of the life form in our models, we incorporate into the hybrid automata "unknown" capacities and a reinforcement variable. The model depends on the presumption that the driver begins at a given speed and has an ideal traveling speed on the interstate. Moving onto the road he attempts to dodge high powers because of increasing speed or direction change, to remain as far to one side as could be expected, and, more than anything, evade crashes with different vehicles. The three conditions of the automata address three unique periods of the move. In methodology, the vehicle cannot yet move off the increasing speed path and just methodologies the circumstance with a steady speed. In path change, the genuine speeding up and path changing happens. In movement, the vehicle again keeps speed and point to the street consistent (Donepudi, 2014a). We utilize this state to assess whether the state toward the finish of the path change behavior is adequately sufficient to save the vehicle on the freeway for an important measure of time.

The vehicle has a position, a current speed, and a point to the path. Those four factors are connected undeniably. The factors speed and path can be constrained by the driver using the capacities, addressing acceleration and steering individually. It is those two capacities we enhance towards maximal reinforcement value. This reinforcement value is addressed by the value into which we incorporate all negative experiences possible while driving onto the freeway. At long last, we model another vehicle driving on the correct path of the expressway through the position and its speed to make the circumstance all the more fascinating.

To get visual feedback about our models, we composed a mathematical test system for our hybrid automata models. It can either estimate the total state-space of the automata or execute (a lot quicker) monte-carlo approximations to appraise the normal value of the reinforcement value. The monte-carlo approach specifically empowers us to utilize hereditary calculations, which need to execute a model ordinarily, to look for those values of the unknown capacities amplifying the normal reinforcement value.

Figure 7 shows the reproduction aftereffects of the model, defining the behavior of the driver after having advanced for the maximal possible reinforcement value. The upper board shows how a vehicle proceeds onward the freeway with a similar vehicle effectively out and about. In the reproduction envisioned in the second board, we raised the speed of vehicle 2 simply a bit and, as a result, noticed a sudden subjective move in the behavior of vehicle 1 conduct. Rather than accelerating and entering before vehicle 2 it moves all the more gradually, enters after vehicle 2, and overtakes it right in the wake of having entered. Note that in no piece of the model we determined a behavior like "overtaking" or "filtering into the surge of traffic". The noticed conduct has arisen normally from the proposed reinforcement values and the streamlining measure.

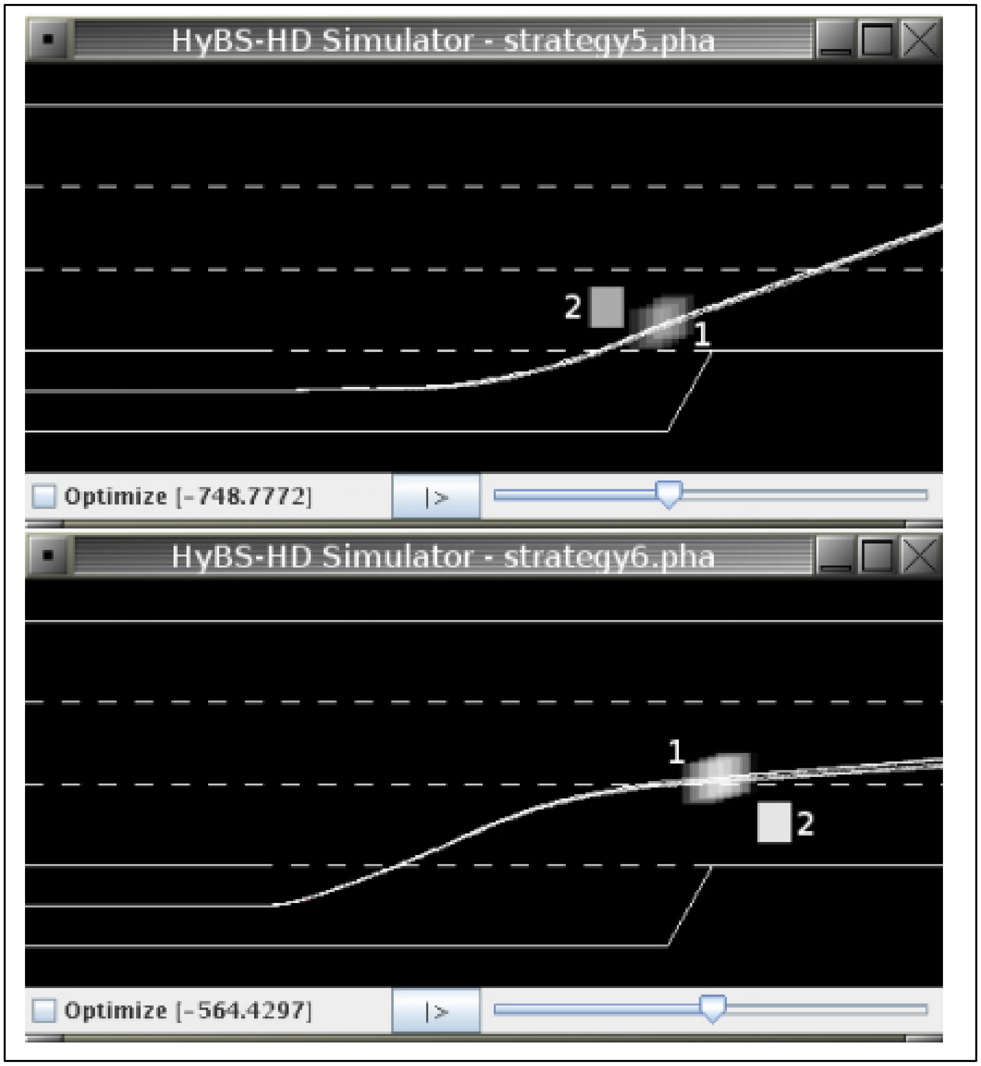
HyBS-HD Simulator - strategy5.pha
2
1
□ Optimize [-748.7772]
|>
HyBS-HD Simulator - strategy6.pha
1
2
□ Optimize [-564.4297]
|>

Figure 7: Simulation of Model

Conclusion

The point of this paper was to give an overview of the capability of hybrid models in the area of behavioral brain research and empirical science as a rule. We discussed the utilization of hybrid automata to display the complex behavior of living beings. In addition to some broad instances of hybrid systems inside the setting of creature behavior, we built a fundamental model of a human driver entering a freeway. The aftereffects of the simulation so far relate well to the conduct one would expect of a genuine driver. On the subjective level, the model predicts "filtering into the traffic" as well as "overtaking" while never having determined any such practices. There is no requirement for a psychological process for starting these practices. All things considered, a basic input learning system like operant conditioning is very much sufficient to clarify even complex driving patterns. We accept this as an illustrative model for the capability of a simple social model of human drivers as far as a hybrid system is concerned. This cycle of displaying unpredictable, heterogeneous systems, is additionally confused by the need to utilize various diverse modeling languages, each intended to work inside an alternate area. Building up a legitimate interface between these modeling languages is significant. In this paper, such an interface was characterized between continuous systems (indicated as common differential conditions) and discrete systems (determined as a limited state machine). Different research studies of such reasonable interest are under analysis and are expected to bloom in the coming time.

References

Donepudi, P. K. (2014a). Technology Growth in Shipping Industry: An Overview. *American Journal of Trade and Policy*, *1*(3), 137-142. https://doi.org/10.18034/ajtp.v1i3.503

Donepudi, P. K. (2014b). Voice Search Technology: An Overview. *Engineering International*, *2*(2), 91-102. https://doi.org/10.18034/ei.v2i2.502

Donepudi, P. K. (2015). Crossing Point of Artificial Intelligence in Cybersecurity. *American Journal of Trade and Policy*, *2*(3), 121-128. https://doi.org/10.18034/ajtp.v2i3.493

Donepudi, P. K. (2017). Machine Learning and Artificial Intelligence in Banking. *Engineering International*, *5*(2), 83-86. https://doi.org/10.18034/ei.v5i2.490

Frehse G. (2005) PHAVer: Algorithmic Verification of Hybrid Systems Past HyTech. In: Morari M., Thiele L. (eds) Hybrid Systems: Computation and Control. HSCC 2005. Lecture Notes in Computer Science, vol 3414. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-31954-2_17

Henzinger T. A., Ho PH., Wong-Toi H. (1997) HyTech: A model checker for hybrid systems. In: Grumberg O. (eds) Computer Aided Verification. CAV 1997. Lecture Notes in Computer Science, vol 1254, 460–463. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-63166-6_48

Möbus C., Eilers M., Garbe H., Zilinski M. (2009) Probabilistic and Empirical Grounded Modeling of Agents in (Partial) Cooperative Traffic Scenarios. In: Duffy V.G. (eds) Digital Human Modeling. ICDHM 2009. Lecture Notes in Computer Science, vol 5620. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02809-0_45

Möhr, M. (1989), Psychobiologie. Grundlagen des Verhaltens. Herausgegeben von K. Immelmann, K. R. Scherer, C. Vogel und P. Schmoock. 888 Seiten, 269 Abb., 34 Tab. Gustav Fischer Verlag, Stuttgart, New York und Psychologie-Verlags-Union, Weinheim, München 1988. Preis: 118,– DM. Nahrung, 33: 472-472. https://doi.org/10.1002/food.19890330520

Salvucci, D. D., Boer, E. R., & Liu, A. (2001). Toward an Integrated Model of Driver Behavior in Cognitive Architecture. Transportation Research Record, 1779(1), 9–16. https://doi.org/10.3141/1779-02

Taher-Uz-Zaman, M., Ahmed, M. S., Hossain, S., Hossain, S., & Jamal, G. R. A. (2014). Multipurpose Tactical Robot. *Engineering International*, *2*(1), 21-27. https://doi.org/10.18034/ei.v2i1.204

--0--